Assigning Telephone Operators to Shifts at New Brunswick Telephone Company

Gary M. Thompson

Cornell University

Abstract

I developed a procedure for assigning telephone operators to shifts at New Brunswick Telephone Company (NBTel). Although the problem has received scant attention in the literature, its solution greatly affects employees' satisfaction with their work schedules. NBTel requires that all shifts be assigned to employees, and it is obligated contractually to satisfy preferences for shifts in order of employee seniority. The specialized shift assignment heuristic (SSAH) that I developed runs on a personal computer, generating approximately three solutions per second. Employee and shift databases are maintained in a spreadsheet, and macros are used to integrate the heuristic into the spreadsheet. Both management and employees see SSAH as an improvement over the previous manual procedure.

Keywords: industries, communications, decision analysis, applications

Assigning Telephone Operators to Shifts at New Brunswick Telephone Company

The Operator Services Department at New Brunswick Telephone Company (NBTel) faced a problem in assigning employees to shifts. The company's objective was to ensure that all shifts would be covered, while satisfying employee choices for shifts as best as possible. Staff performed this procedure manually, using a spreadsheet as a support tool.

Although a substantial amount of literature exists on the problem of workforce scheduling [for example, Agnihothri and Taylor 1991, Brusco et al. 1995, Love and Hoey 1990, Schindler and Semmel 1993, Thompson 1990, 1995, and 1996], little direct work has been done on problems like the one facing NBTel. I have characterized the labor scheduling problem as having four steps [Thompson 1995]: (1) forecast customer demand; (2) translate the forecasts into requirements for employees; (3) develop the labor schedule; and (4) control the delivery of the service in real time. NBTel's problem is an often-ignored component of step three in this typology, specifically assigning individual employees to the work schedules. The argument in favor of ignoring this task was that full-time employees have to work whenever customer demand warrants. Some research on part-time employees has captured the essence of the task by scheduling employees for work (in step two) only during the portion of the operating day they are available [Thompson 1990] or want to work [Love and Hoey 1990]. However, the issue of satisfying the desires of full-time employees remained unaddressed and is of particular concern at NBTel, since their contractual agreement requires that the company satisfies employee requests for shift choices if possible.

The Shift Assignment Problem (SAP)

There are several inputs to the shift assignment problem (SAP). First, every week, schedulers identify the shifts that will be needed to satisfy expected customer demand for the second week out. In practice, they use a set of basic shifts to which they add supplementary shifts as needed. The shifts are categorized into 11 types, based on the skills they require and the location at which they can be performed.

The second input is the availability of the employees. The employees receive alternate weekends off, and they may also request specific days off for the week being scheduled. Employees assigned tasks outside the "pool" for a portion of the work are shown as unavailable during those times, and the maximum number of shifts they should be assigned is adjusted downward appropriately. A typical schedule has approximately 70 to 90 employees working approximately 40 daily shifts, on average, and contains 90 to 100 unique shifts (appendix).

Every quarter, employees bid for shifts. In this bidding, each employee specifies his or her ideal work assignment (typically the particular shifts they want to work). Previously, the staff assigned shifts manually, attempting to ensure that they satisfied the employees' shift preferences. Contractually, NBTel was obligated to satisfy the preferences in order of employee seniority.

The way NBTel incorporates employee preferences has certain advantages over letting the employees choose shifts sequentially in order of their seniority. Both methods allow employees to get shifts that start at or close to the times they desire. The latter method, however, can result in some leftover shifts that the least senior employees cannot work, either because they are not available or because they do not have the necessary skills. The latter method can also miss good compromises. For example, assigning a more senior employee a day off on Tuesday instead of Wednesday might have a ripple effect that improves the satisfaction of many less senior employees. This trade-off is good because the employees are generally indifferent to which weekday off-days they are assigned. If a particular employee requests a specific day off, he or she is simply recorded as unavailable on the day.

The SAP can be thought of as a goal programming problem with many objectives. In modeling it, the highest (preemptive) priority is to assign as many of the shifts as possible. A complete assignment of shifts is not always possible, because of the number and characteristics of the shifts and the number, location, and skills of the available employees. Subsequent preemptive priorities (and there is one for each employee) attempt to satisfy the employees' desired shift choices in the order of their seniority. For example, the third most senior employee will receive his or her desired shifts, so long as this does not affect the satisfaction of the two most senior employees. For seniority purposes, full-time and part-time employees are treated identically.

The constraints that are present ensure that employees are assigned only to shifts for which they are available and have the necessary skills; that all employees receive at least one shift per week; that an employee cannot be assigned more than one shift per day; and that a senior employee receives five (or his or her desired maximum number of) shifts before a less senior employee receives more than one shift (a contractual restriction). The appendix contains the mathematical form of the SAP.

General Dissatisfaction with Existing Manual Procedure

Management was dissatisfied with the manual procedure for assigning shifts because it took a great deal of time to perform it. The employees were dissatisfied, because many felt that some less senior employees often received better shifts. As a result of this dissatisfaction, NBTel brought me in to search for or develop an automated decision support aid.

Because the company had experience with spreadsheets, I looked for an approach that would permit the scheduling staff to maintain the necessary database of shifts, employee preferences for shifts, and other relevant data on a spreadsheet. The SAP is a binary goal-programming (GP) problem with many objective components, variables, and constraints (appendix). The need for specialized software, substantial RAM, and sizable run time negated the possibility of searching for an optimal solution using commercial GP software. I settled on developing a specialized shift assignment heuristic (SSAH) for solving the SAP that could run on a standard 486- based PC.

SSAH

SSAH is a stand-alone procedure for solving the SAP that is used in conjunction with spreadsheet macros. The macros write a description of the problem to a file, which SSAH subsequently retrieves. The spreadsheet macros also perform some simple checks on the data. For example, one macro determines if too many shifts have been scheduled for the weekend days given the number of employees who should receive these days off. In the initial development of a weekly schedule, the schedulers run SSAH for a few minutes, during which time they will get a good indication of whether the shifts can all be assigned. After making changes to the schedule, such as adding employees to the pool, they run SSAH for 10 to 15 minutes to obtain the solution that is implemented.

Here is the algorithm SSAH follows:

A. INITIALIZE-Perform preliminary calculations and clear any shift assignments.

B. ASSIGN-Assign shifts to employees in order of seniority.

C. If all shifts have not been assigned, use MAKEFEAS-Search for two- and three-way shift swaps that assign unassigned shifts.

D. If all shifts have been assigned or if the number of unassigned shifts equals the fewest number of unassigned shifts yet found, then:

D.1 IMPROVE-Evaluate all two-way shift swaps between employee pairs, searching for superior assignments.

D.2 If the current assignment is the best found, record it.

E. While time remains, repeats steps A-D; otherwise write the best assignment to a file and terminate SSAH.

SSAH consists of several routines that are used iteratively until the available time has expired. Each iteration begins with INITIALIZE. INITIALIZE calculates the number of shifts each employee will be assigned, identifies which day I shift-type combinations will likely cause staffing problems, and, for subsequent iterations, clears the shift assignments. Next, ASSIGN assigns shifts to employees. This assignment is done in the order of employee seniority and incorporates a random selection of equally desirable shifts, so that each iteration of SSAH typically yields a different final solution. As shifts are assigned, the set of unassigned shifts becomes smaller. Without the look-ahead features that it incorporates in ASSIGN, SSAH could reach a point where the least senior employees could not work any of the unassigned shifts (in general, the least-senior employees can work the fewest types of shifts). This difficulty can also arise when employees are allowed to pick their shifts in order of seniority. SSAH's look-ahead feature forces the assignment of less-than-ideal shifts to more senior employees if doing so results in fewer unassigned shifts. Thus, the look-ahead feature imposes the highest priority objective of the SAP-minimizing the number of unassigned shifts-at the expense of subsequent priorities regarding employee preferences.

SSAH allows employee preferences to be specified by a desired starting time, a desired lunch length (some shifts have 30-minute lunches, while others have 60-minute), and measures of the undesirability of earlier- or later-than-desired shift starts. Specifying preferences in this form is very useful for less-senior employees who are unlikely to be assigned their ideal shifts. As an example of the preference format, consider an employee who wants to start work at 7:00 AM, but if he cannot be assigned a 7:00 AM shift, he prefers a 7:30 AM shift to a 6:30 AM shift, but prefers a 6:30 AM shift to an 8:00 AM shift. These preferences could be represented as undesirability indices of 1.5 for early starts and 1.0 for late starts.

If, after ASSIGN has assigned as many shifts as possible, there are still some unassigned shifts, MAKEFEAS is used to attempt to assign the remaining shifts. MAKEFEAS searches for two- and three-way shift swaps that assign the unassigned shifts. In a three-way swap, employee A might be assigned to an unassigned shift, employee B to the shift vacated by employee A, and employee C, who was short (had been assigned an insufficient number of shifts), would be assigned to the shift originally assigned to employee A.

After MAKEFEAS, if the assignment is feasible (all shifts are assigned) or the number of unassigned shifts equals the smallest yet found, IMPROVE is used to try to find an improved assignment. IMPROVE evaluates all two-way shift swaps between all employee pairs. A swap occurs if it improves the satisfaction of the more senior employee in the pair.

If the resultant assignment is the best yet found, it is recorded. Then, if run time remains, SSAH begins redeveloping a new assignment from scratch. In the initial development of a weekly schedule, schedulers let SSAH run for a few minutes, during which time they will get a good indication of whether the shifts can all be assigned. After making changes to the schedule, such as adding employees to the pool, they run SSAH for 10 to 15 minutes to obtain the solution that is implemented. After the specified run time has expired, SSAH writes the best assignment to a file, which is subsequently loaded into the spreadsheet for manipulation and printing.

A reasonable question is why a new assignment is developed from scratch in each iteration instead of using a local search heuristic to make improvements. I first tried using a simulated annealing-based procedure in which changes were made to the assignment by swapping randomly selected compatible shifts between randomly selected employees (in a manner based on what I had done for another labor-scheduling problem [Thompson 1996]). The difficulty with this approach is that these swaps are much more likely to make the solution worse than they are to make it better. Basically, the solution deteriorates as more iterations are run, due to the many preemptive objective priorities. Another approach I tried was to drop a randomly selected number of shifts from each employee (yielding a partial assignment) and then redevelop the assignment from that starting point. This also proved inferior to a complete reassignment, because the variation in final assignments derived from these partial assignments was much smaller than the variation in final assignments developed from scratch.

For the typical SAP, SSAH generates approximately three complete assignments per second on a Pentium 90-based personal computer. NBTel usually runs SSAH on 486-based PC.

The Outcome

Both management and employees have expressed satisfaction with SSAH. The employees have "bought in" to its ability to make fair assignments. In fact, it brought about an unexpected change in the process of employees' specifying their desired shift choices. Where previously most of the more senior employees specified particular shifts, they now specify their desired start times, lunch lengths, and their undesirability indices for early and late starts. This has resulted in a much more streamlined shift-bidding process. Previously, the bidding process was time-consuming because each employee had to bid in order of seniority. If a person was out sick, the bidding process would be delayed. Now, the employees know that they just have to specify their desired shift characteristics, and SSAH will ensure that shift assignments observe seniority rules. Finally, the employees welcome SSAH's ability to handle their requests for specific weekdays off.

Management has been equally pleased, because SSAH has eliminated one source of employee discontent and because the staff can easily perform what-if analyses regarding the assignment. For example, they can now easily evaluate the effect on satisfaction of increasing cross-training, the effect of an employee wanting a specific day off, or the effect of an employee altering his or her desired shift characteristics. The success with the procedure has resulted in other departments expressing interest in using SSAH.

References

Agnihothri, Saligrama A. and Taylor, Patricia F. 1991, "Staffing a centralized appointment scheduling department in Lourdes Hospital," Interfaces, Vol. 21, No. 5 (September-October), pp. 1-11.

Brusco, Michael J.; Jacobs, Larry W.; Bongiorno, R. J.; Lyons, D. V.; and Tang, B. 1995, "Improving personnel scheduling at airline stations," Operations Research, Vol. 43, No. 5 (September-October), pp. 741- 751.

Love, Robert R., Jr. and Hoey, James M. 1990, "Management science improves fast food operations," Interfaces, Vol. 20, No. 2 (MarchApril), pp. 21-29.

Schindler, Sol and Semmel, Ted 1993, "Station staffing at Pan American World Airways," Interfaces, Vol. 23, No. 3 (May-June), pp. 91-106.

Thompson, Gary M. 1990, "Shift scheduling in services when employees have limited availability: An LP approach," Journal of Operations Management, Vol. 9, No. 3 (August), pp. 352- 370.

Thompson, Gary M. 1995, "Labor scheduling using NPV estimates of the marginal benefit of additional labor capacity," Journal of Operations Management, Vol. 13, No. 1 Ouly), pp. 67- 86.

Thompson, Gary M. 1996, "A simulated annealing heuristic for shift scheduling using non-continuously available employees," Computers and Operations Research, Vol. 23, No. 3 (March), pp. 275-288.

APPENDIX

The SAP

The mathematical model describing NBTel's shift assignment problem, SAP, consists of the following:

$S =$ shifts;

$E =$ employees, ordered from most senior to least senior;

$C_e =$ shift categories that employee e can work (defined in practice by the employees' skills and location);

$D_e =$ days that employee e can work.

Decision Variables (Binary)

$$x_{es} = \begin{cases} 1, \text{if employee } e \text{ is assigned to shift } s \\ 0, \text{otherwise;} \end{cases}$$

$$y_e = \begin{cases} 1, \text{if emplyee } e \text{ is assigned the maximum number of shifts} \\ 0. \text{otherwise;} \end{cases}$$

$$u_s = \begin{cases} 1, \text{if shift } s \text{ is unassigned} \\ 0. \text{otherwise.} \end{cases}$$

Parameters

$m_e =$ the maximum number of shifts to assign to employee e;

$v_{es} =$ the undesirability of assigning employee e to shift s (defined by employee e);

$c_s =$ the category of shift s;

$d_s =$ the day on which shift s is scheduled.

Objective Function

The objective is a set of preemptive priorities. The first priority is to minimize the number of unassigned shifts. Subsequent priorities specify that the employees' shift choices are to be satisfied, if possible, in order of seniority.

$$\text{Min } Z = P_0[\textstyle\sum_{s\in S} u_s] \tag{1}$$

$$+\sum_{e\in E} P_e\left(\sum_{\{s\in S|c_s\in C_e,d_s\in D_e\}} v_{es}x_{es}\right)$$

where $P_i \gg P_{i+1}$ (the priority for component *i* is very much greater than the priority for component i + 1).

Constraints

Assign all shifts

$$\sum_{\{e\in E|c_s\in C_e,d_s\in D_e\}} x_{es} + u_s = 1 \text{ for } s \in S. \tag{2}$$

At least one shift to each employee

$$\sum_{\{s\in S|c_s\in C_e,d_s\in D_e\}} x_{es} \geq 1 \text{ for } e \in E. \tag{3}$$

Assign no more than the maximum desired number of shifts for each employee

$$\sum_{\{s\in S|c_s\in C_s\in C_e,d_s\in D_e\}} x_{es} \leq m_e \text{ for } e \in E \tag{4}$$

No more than one shift per day per employee

$$\sum_{\{s\in S|c\_s\in C\_e,d\_s=i\}} x_{es} \leq 1\ for\ e \in E, i \in D_e. \tag{5}$$

Force a more senior employee to work their desired number of shifts before a less senior employee works more than one shift

$$\sum_{\{s \in S | c_s \in C_e, d_s \in D_e\}} x_{es}$$

(6)

$$m_e y_e \leq \sum_{\{s \in S | c_s \in C_e, d_s \in D_e} x_{es}$$

(7)

for $\{ e \in E \mid e < E \}$.

In practice, it is a simple matter to determine the number of shifts each employee will be assigned after first identifying the total number of shifts to be assigned. There will be an employee, say *j*, such that all the more senior employees will work their desired number of shifts, all less senior employees will work one shift, and employee *j* will work at least one, but no more than m_j shifts. Thus, constraints (6) and (7) can be replaced with

(8)

$$\sum_{\{s \in S | c_s \in C_e, d_s \in D_t\}} X_{es}$$

$$= m_e \text{ for } \{ e \in E \mid e < j \},$$

$$\sum_{\{s \in S | c_s \in C_j, d_s \in D_e\}} x_{js} < m_j, \text{and}$$

(9)

$$\sum_{\{s \in S | c_s \in C_e, d_s \in D_e} x_{es}$$

(10)

Finally, there are restrictions that impose the binary nature of the variables:

(11)

$$x_{es} = \{0,1\} \text{ for } e \in E,$$

$$\{s \in S \mid c_s \in C_e, d_s \in D_e\}$$

$$y_e = \{0,1\} \text{ for } \{e \in E \mid e < E\}$$

(12)

$$u_s = \{0,1\} \text{ for } s \in S$$

(13)

An Example

Table 1 contains data on the 73 employees available: their desired shift characteristics, the skills they have, the maximum number of shifts they can work, and the days on which they should not be assigned. Table 2 lists the specific shifts that must be staffed on each day. Table 3 contains the characteristics of the 96 unique shifts (some shifts are scheduled on several days): each unique shift' s start time, lunch length, and required skill. Finally, Table 4 shows the shift assignment I developed using SSAH (on a Pentium 90 with a run time of five minutes). It identifies the specific shifts each employee is working, which days he or she is not working, and his or her overall satisfaction index. A satisfaction index of zero indicates perfect satisfaction.

Consider, for illustration, employee 1, and why this employee is not perfectly satisfied. Employee 1 can be assigned only shifts requiring skill type 9. Only one shift requiring skill type 9 is in the schedule – shift 95, which is scheduled every day of the week. Since employee 1 desires a start time of period 25, but shift 96 starts in period 24, employee 1 will start one period earlier than desired on each of the five days she works. Since the employee's undesirability of an early start is 3, 3 undesirability units per period per shift*l period*5 shifts gives a total satisfaction index of 15.

New Brunswick Telephone Company

Box 1430, Saint John, New Brunswick E2L 41<2, Canada, writes, "We feel the software now addresses virtually all the concerns we had and has successfully dealt with not only the contractual concerns but also the more obscure 'local constraints' that had grown up over the years in a manual assignment environment and to the staff are as important as those in the contract.

"After a fair trial period, the staff have voted to adopt the system and so replace the very cumbersome and arbitrary manual process we have used for many years with all its shortcomings and dissatisfiers.

"The most common viewpoint the staff express is that the system is 'more fair' in distributing the shifts and adhering to seniority in doing so. The flexibility in stating preferences is also proving popular, i.e., biasing the system to earlier or later shifts."

Table 1. In this table of employee shift preference and availability data for a typical week in the operator services department at NBTel, EMP is the employee number (in order of seniority), ST is the desired start time (1 = 6:00 am, 2 = 6:30 am, etc.), LL is the desired lunch break length (in half hours), ES is the undesirability index for an early start or an early finish, LS is the undesirability index for a late start or late finish, SklTyp is the skills of the employee, MxWk is the maximum number of shifts the employee is to be assigned, and UnDay is the days that the employee should not be scheduled for work.

EMP	ST	LL	ES	LS	SklTyp	MxWk	UnDay	EMP	ST	LL	ES	LS	SklTyp	MxWk	UnDay
E-01	25		3		9	5		E-38	22			3	1, 4, 5, 10, 11	5	Sun, Tue
E-02	24		2		5	5	Mon, Sat	E-39	21		4		4, 5, 10, 11	5	Sun, Fri
E-03	4		2		2	5		E-40	37		4	1	6, 7	5	Sun, Fri
E-04	3		3	1	1	5	Sun, Fri	E-41	7		1	3	1, 2	5	Tue, Sat
E-05	4		1	3	1, 8	5	Fri, Sat	E-42	19		2		1, 2	5	Sun, Wed
E-06	5		3	1		5	Sun, Mon	E-43	26		3		1, 2	5	Sun, Fri
E-07	3			3		5	Mon, Sat	E-44	4				1, 2, 8	5	Sun, Wed
E-08	5			2	1, 8	5	Fri, Sat	E-45	7		2		1, 2	5	Sun, Tue
E-09	37		4	1	1, 7	5	Wed, Thu	E-46	37		4		6	3	
E-10	3		1	3	1	5		E-47	16		4		1, 2, 6	2	
E-11				3		5		E-48	19		2	1	1, 2, 6	2	
E-12	4	2			1, 3, 10	5		E-49	19		4	2	1, 2, 6	3	
E-13				3	1, 8	5	Sun, Tue	E-50	19		1	3	1, 2	2	
E-14	5		2		1, 8	5	Sun, Wed	E-51	21		4		1, 2	3	
E-15	7			2	1, 3	5		E-52	21		4		1, 2	4	
E-16	20		2		1	5		E-53	19				1, 2	5	Tue, Wed
E-17	19		12			5		E-54	19		8		1, 2	3	
E-18	4		2		1, 8	5	Thu, Sat	E-55.	22		4		1, 2	2	
E-19	19		3			5	Fri, Sat	E-56	18		2.7		1, 2	2	
E-20	6			6	1, 2	5	Sun, Fri	E-57	25		2		1, 2	2	
E-21	19		12	1	1	5		E-58	18		6		l, 2	2	
E-22	4			3	1, 2	5		E-59	18		2.7	1	1, 2	3	
E-23	28		3	1	1	5	Sun, Mon	E-60	5			3	1, 2	2	
E-24	20		1	1.5	1, 2	5	Sun, Wed	E-61	3			3	1, 2, 4	5	Sun, Sat
E-25	4	1		3	1, 2, 3, 4, 10	5	Sun, Thu	E-62	5				1, 2	5	Sun, Sat
E-26	6	2		2	1	5	Mon, Sat	E-63	7				1, 2	5	Sun, Sat
E-27	4		1.5		l, 2	5	Sun, Thu	E-64	5		1		1, 2	5	Tue, Sat

E-28	21	2		1, 2	5	Sun, Tue	E-65	15	4		1, 2	3	
E-29	23	1	3	1, 9	5	Sun, Wed	E-66	4	2		1, 2	5	Sun, Sat
E-30	19	2		1	5		E-67	15	4		1, 2	2	
E-31	24		1		5	Mon, Sat	E-68	21	4		1, 2	2	
E-32	7		3		5	Wed, Sat	E-69	17	4		1, 2	3	
E-33	19		1	1, 2	5		E-70	21	4		1, 2	2	
E-34	37	1	8	6, 7	5	Mon, Sat	E-71	13	4		1, 2	2	
E-35	20	2	1	1, 5, 9	5	Sun, Tue	E-72	21	4	1	1, 2	2	
E-36	19	2.7	1	1, 2	5	Mon, Thu	E-73	5		2	1, 2	5	Thu
E-37	11	3		1, 2	5								

Table 2. Shift requirements vary by day for a typical week in the operator services department at NBTel

Day	Specific Shifts That Must be Shifted	Total Number of Shifts
Sunday	62-86, 92-96	30
Monday	1-36, 88-92, 94-96	44
Tuesday	1-36, 88-92, 94-96	44
Wednesday	1-36, 88-92, 94-96	44
Thursday	1-36, 88-92, 94-96	44
Friday	1-36, 87-92, 94-96	45
Saturday	37-61, 87, 92-96	31

Table 3. In this table of shift characteristics for a typical week in the operator services department at NBTel, SN is the shift number, ST is the shift start time (1 = 6:00 am, 2 = 6:30 am, etc.), BL shift's lunch length, in half hours, and TT is the skill required of an employee working the shift.

SN	ST	BL	TT	SN	ST	BL	TT	SN	ST	BL	TT	SN	ST	BL	TT
1	37	1	6	25	19	1	1	49	10	2	1	73	9	1	1
2	37	1	6	26	19	1	1	50	11	2	1	74	11	2	1
3	3	1	1	27	20	1	1	51	13	2	1	75	11	2	1
4	4	1	1	28	20	1	1	52	20	1	1	76	13	2	1
5	4	1	2	29	23	1	1	53	20	1	1	77	19	1	1
6	4	2	2	30	23	1	1	54	21	1	1	78	21	1	1
7	5	1	1	31	24	1	1	55	22	1	1	79	21	1	1
8	5	2	1	32	24	1	1	56	24	1	1	80	21	1	1
9	6	1	1	33	24	1	1	57	25	1	1	81	22	1	1
10	6	1	1	34	25	1	2	58	25	1	2	82	23	1	1
11	6	2	1	35	25	1	1	59	25	1	1	83	23	1	1
12	7	1	1	36	28	1	1	60	27	1	1	84	24	1	1
13	7	1	1	37	37	1	6	61	28	1	1	85	25	1	2
14	7	1	1	38	37	1	6	62	37	1	6	86	26	1	1
15	7	1	1	39	4	1	2	63	37	1	6	87	28	1	1
16	7	1	1	40	4	2	1	64	37	1	6	88	4	2	4
17	7	2	1	41	5	1	1	65	4	1	2	89	5	2	3
18	8	1	1	42	6	1	1	66	4	2	1	90	7	2	4
19	8	1	1	43	6	1	1	67	6	1	1	91	22	1	11
20	10	2	1	44	7	1	1	68	6	1	1	92	24	1	5
21	9	1	1	45	7	1	1	69	7	1	1	93	4	2	10
22	11	2	1	46	7	2	1	70	7	1	1	94	4	2	8
23	19	1	1	47	8	1	1	71	7	1	1	95	24	1	9
24	19	1	1	48	9	6	1	72	8	1	1	96	37	1	7

Table 4. In the staffing assignment generated by SSAH for a typical week in the operator services department at NBTel, EMP is the employee number (in order of seniority) and SI is the total satisfaction index for the employee (a value of zero indicates perfect satisfaction). Values in the daily columns are the shifts that the employee works. NA indicates that the employee was not to assigned on a particular day.

EMP	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	EMP	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
E-01	95		95	95	95		95	15	E-38	NA	91	NA	91	91	91	51	8.5
E-02	92	NA	92	92	92	92	NA	0	E-39	NA	92	91	90	90	NA	93	178
E-03	64		5	5		5	39	0	E-40	NA	2	2	2	2	NA	87	0
E-04	NA	3	3	3	3	NA	40	1.5	E-41	75	16	NA	16	12	15	NA	20
E-05	65	4	4	4	4	NA	NA	1.5	E-42	NA	33	33	NA	31	32	59	26
E-06	NA	NA	7	7	7	7	41	0	E-43	NA	36	35	35	35	NA	60	12
E-07	67	NA	8	8	8	3	NA	32	E-44	NA	14	15	NA	16	13	48	20
E-08	94	7	94	94	94	NA	NA	2	E-45	NA	17	NA	12	15	17	58	19
E-09	0	0	0	NA	NA	0	0	0	E-46	63					1	37	0
E-10	66	8	10			4	43	38	E-47	84						38	30
E-11	68	10	9	10		8		77	E-48	85					87		25
E-12	93	89	89		89	89		4	E-49		1		1	1			108
E-13	NA	94	NA	9	10	94	94	62	E-50			22	22				15
E-14	NA	9	11	NA	9	9	42	5.5	E-51		32	31			31		9
E-15	70	12		14		12	44	0	E-52		31	34	33	33			13
E		*V*	*V*					1	3		35	NA	NA	34	34	57	33
E-17	76		26	26	24	23		0	E-54		34		31		35		17
E-18	69	11	13	11	NA	10	NA	13	E-55				34		20		49
E-19	77	25	24	23	25	NA	NA	2	E-56		20		20				40
E-20	NA	6	6	6	6	NA	45	12	E-57			20		20			58
E-21	79	23	23	24		24		2	E-58				21		21		108
E-22	71	5			5	6	46	24	E-59		21	21		21			72
E-23	NA	NA	36	36	36	36	61	0	E-60				17	14			14
E-24	NA			NA	27	28	53	0	E-61	NA	90	90	88	88	90	NA	50
E-25	NA	88	88	89	NA	88	47	21	E-62	NA		17				NA	2.5
E-26	72	NA	12	15	11	11	NA	7	E-63	NA				17		NA	0.5
E-27	NA	13	14	13	NA	16	49	19	E-64			NA	19			NA	3
E-28	NA	29	NA	29	30	29	54	8	E-65						18		
E-29	NA	30	29	NA	29	30	55	1	E-66	NA			18			NA	4
E-30	80	24		25	26	25		3	E-67						19		
E-31	83	NA	32	32	32	33	NA	0	E-68			18					52
E-32	73	15	16	NA	13	14	NA	14	E-69					19			36
E-33	82	26	25			26	52	5	E-70		18						52

E-34	62	NA	1	0	0	2	NA	0	E-71			19					20
E-35	NA	95	NA	27	23	95	92	14	E-72					18			52
E-36	81	NA	30	30	NA	27	56	18	E-73		19			NA			6
E-37	74	22			22	22	50	2.5									